

# Answer 1

Console Output:

```
ubuntu@ip-172-31-86-102:/$ curl https://www.wetransfer.com -v
* Trying 52.213.142.26:443...
* Connected to www.wetransfer.com (52.213.142.26) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* CAfile: /etc/ssl/certs/ca-certificates.crt
* CAPath: /etc/ssl/certs
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS header, Certificate Status (22):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS header, Finished (20):
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS header, Certificate Status (22):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES128-GCM-SHA256
* ALPN, server accepted to use h2
* Server certificate:
* subject: CN=wetransfer.com
* start date: Jul 7 00:00:00 2022 GMT
* expire date: Aug 5 23:59:59 2023 GMT
* subjectAltName: host "www.wetransfer.com" matched cert's "*.wetransfer.com"
* issuer: C=US; O=Amazon; OU=Server CA 1B; CN=Amazon
* SSL certificate verify ok.
* Using HTTP2, server supports multiplexing
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* Using Stream ID: 1 (easy handle 0x55e8e2bcca50)
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
> GET / HTTP/2
> Host: www.wetransfer.com
> user-agent: curl/7.81.0
> accept: */*
>
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* Connection state changed (MAX_CONCURRENT_STREAMS == 128)!
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
< HTTP/2 301
< date: Fri, 15 Jul 2022 15:45:18 GMT
< content-length: 0
< location: https://wetransfer.com/
<
* Connection #0 to host www.wetransfer.com left intact
ubuntu@ip-172-31-86-102:/$
```

1. We start off with the command itself. It is used to make HTTP & HTTPS requests to an endpoint. Curl more commonly used in unix systems supports HTTP, HTTPS, FTP, SCP and many more protocols.
2. The command get picked up and interpreted by the shell, it changes if the keywords being used is a program contained in the shell itself.
3. The built-in program of curl which is present in /bin/curl (taken from \$PATH/curl) is executed and passed on to the kernel to initiate communication with the NIC. (Network Interface Card)

```

ubuntu@ip-172-31-86-102:/bin$ pwd
/bin
ubuntu@ip-172-31-86-102:/bin$ ls -la | grep curl
-rwxr-xr-x 1 root root 260328 May  9 12:34 curl
ubuntu@ip-172-31-86-102:/bin$

```

```

ubuntu@ip-172-31-86-102:/bin$ printenv
SHELL=/bin/bash
PWD=/
LOGNAME=ubuntu
XDG_SESSION_TYPE=ttty
MOTD_SHOWN=pam
HOME=/home/ubuntu
LANG=C.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=
az=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31
.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31
31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.
5:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2V
*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;
;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.og
SSH_CONNECTION=122.161.53.217 6262 172.31.86.102 22
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm
LESSOPEN=| /usr/bin/lesspipe %s
USER=ubuntu
DISPLAY=localhost:10.0
SHLVL=1
XDG_SESSION_ID=1
XDG_RUNTIME_DIR=/run/user/1000
SSH_CLIENT=122.161.53.217 6262 22
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
SSH_TTY=/dev/pts/0
_=/usr/bin/printenv
OLDPWD=/bin
ubuntu@ip-172-31-86-102:/bin$

```

4. The Kernel initiates the communication with NIC through its firmware/drivers and accordingly asks it to perform the tasks.
5. The NIC initiates a UDP call on port 53 to the first DNS server via the router, which usually is your Internet Service Provider or a 3rd Party DNS provider.
6. The host IP Address allocation and the IP address of the DNS resolver is usually configured by the router via DHCP calls when the system connects to the router.
7. Usually the DNS entries for commonly used websites will be found at the first DNS checkpoint (DNS Resolver) itself. Incase the entry is not found then the query is forwarded to the root DNS server, which is in turn forwarded to the .com TLD server which is again forwarded to the domain nameserver which finally returns the IP Address of wetransfer.com. *(In our case, the DNS lookup answer wasn't cached, so our request went all the way to the domain nameserver of wetransfer.com and dns records were retrieved from there, in form of an authoritative response)*

**serial** is a revision numbering system for DNS entries, whenever there are any changes, the serial number changes, and secondary nameservers are alerted

**refresh** specifies the interval after while secondary DNS server will poll the primary one if there is any change in the dns record

**retry** specifies the time length secondary server will wait if primary server is non-responsive.

**expire** specifies the time that secondary DNS server will keep cached DNS records valid for, if the primary server doesn't respond after that then the secondary server will stop responding for such queries

```
> set q=soa
> https://www.wetransfer.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
*** Can't find https://www.wetransfer.com: No answer

Authoritative answers can be found from:
wetransfer.com
    origin = ns-1495.awsdns-58.org
    mail addr = awsdns-hostmaster.amazon.com
    serial = 1
    refresh = 7200
    retry = 900
    expire = 1209600
    minimum = 86400
> █
```

```

ubuntu@ip-172-31-86-102:/bin$ nslookup
> https://www.wetransfer.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   https://www.wetransfer.com
Address: 52.213.142.26
Name:   https://www.wetransfer.com
Address: 52.214.128.28
Name:   https://www.wetransfer.com
Address: 34.250.122.178
>

```

(Note the highlighted IP, 52.214.128.28)

8. The communication starts between the client (curl call from the terminal) and the server. The protocol used will be TCP over port 443 (HTTPS) and it will be a GET request.
9. After establishing the TCP connection and agreeing on the cipher to exchange data, there is a negotiation between the client and the server which is the SSL/TLS handshake.
10. The client sends a hello message packet to the server which contains all the supported ciphers and SSL version. The server then replies with a hello packet of itself, which includes data like the agreed upon cipher to exchange data and SSL version.

#### Client Hello Packet showing all the list of supported ciphers

35	3.613935	192.168.1.68	52.214.128.28	TLSv1.2	571 Client Hello
36	3.793133	52.214.128.28	192.168.1.68	TCP	54 443 → 50962 [ACK] Seq=1 Ack=518 Win=28160 Len=0
37	3.793967	52.214.128.28	192.168.1.68	TLSv1.2	1466 Server Hello
38	3.794151	52.214.128.28	192.168.1.68	TLSv1.2	1466 Ignored Unknown Record
39	3.794225	192.168.1.68	52.214.128.28	TCP	54 50962 → 443 [ACK] Seq=518 Ack=2825 Win=131072 Len=0
40	3.794283	52.214.128.28	192.168.1.68	TLSv1.2	1466 Ignored Unknown Record
41	3.794422	52.214.128.28	192.168.1.68	TLSv1.2	1198 Ignored Unknown Record

```

nsport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 512
  ✓ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 508
    Version: TLS 1.2 (0x0303)
    > Random: 3970a96672eaa843c2e774f362653e5a2fa8889813a29a49a3de605d223e4162
    Session ID Length: 32
    Session ID: a7967a6e9b7cbe44ac4db27814e129c68d494d8611f817d38d385305e06e2142
    Cipher Suites Length: 32
    ✓ Cipher Suites (16 suites)
      Cipher Suite: Reserved (GREASE) (0x5a5a)
      Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
      Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
      Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc03a)
      Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc03b)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
      Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Methods Length: 1
    > Compression Methods (1 method)

```

## Server Hello Packet highlighting the cipher and the TLS version used

36	3.793133	52.214.128.28	192.168.1.68	TCP	54 443 → 50962 [ACK] Seq=1
37	3.793967	52.214.128.28	192.168.1.68	TLSv1.2	1466 Server Hello
38	3.794151	52.214.128.28	192.168.1.68	TLSv1.2	1466 Ignored Unknown Record
39	3.794225	192.168.1.68	52.214.128.28	TCP	54 50962 → 443 [ACK] Seq=5
40	3.794283	52.214.128.28	192.168.1.68	TLSv1.2	1466 Ignored Unknown Record
41	3.794432	52.214.128.28	192.168.1.68	TLSv1.2	1466 Ignored Unknown Record

- > Frame 37: 1466 bytes on wire (11728 bits), 1466 bytes captured (11728 bits) on interface \Device\NPF
- > Ethernet II, Src: Cambridg\_d4:8a:30 (a8:25:eb:d4:8a:30), Dst: AzureWav\_e3:71:43 (ec:2e:98:e3:71:43)
- > Internet Protocol Version 4, Src: 52.214.128.28, Dst: 192.168.1.68
- > Transmission Control Protocol, Src Port: 443, Dst Port: 50962, Seq: 1, Ack: 518, Len: 1412
- ▼ Transport Layer Security
  - ▼ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
    - Content Type: Handshake (22)
    - Version: TLS 1.2 (0x0303)
    - Length: 70
  - ▼ Handshake Protocol: Server Hello
    - Handshake Type: Server Hello (2)
    - Length: 66
    - Version: TLS 1.2 (0x0303)
    - > Random: e6dcc48d6e364b81608989c061c8a53a8b26e0d4d877c69bcf0434a011b3c7fe
    - Session ID Length: 0
    - Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0xc02f)
    - Compression Method: null (0)
    - Extensions Length: 26
    - > Extension: renegotiation\_info (len=1)
    - > Extension: ec\_point\_formats (len=4)
    - > Extension: session\_ticket (len=0)
    - > Extension: application\_layer\_protocol\_negotiation (len=5)
    - [JA3S Fullstring: 771,49199,65281-11-35-16]
    - [JA3S: 8d2a028aa94425f76ced7826b1f39039]

11. Client ensures that it is talking to the right server, by checking the SSL Certificate that is returned by the server which contains the domain name, expiry date and the public key.

43	3.796431	192.168.1.68	52.214.128.28	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
44	3.974640	52.214.128.28	192.168.1.68	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
45	3.974851	52.214.128.28	192.168.1.68	TLSv1.2	123	Application Data
46	3.974906	192.168.1.68	52.214.128.28	TCP	54	50962 → 443 [ACK] Seq=644 Ack=5708 Win=130816 Len=0

Frame 43: 180 bytes on wire (1440 bits), 180 bytes captured (1440 bits) on interface \Device\NPF\_{F73D8206-7163-472D-AD5D-8D75AAE4C7C8}, id 0  
Ethernet II, Src: AzureWav\_e3:71:43 (ec:2e:98:e3:71:43), Dst: Cambridg\_d4:8a:30 (a8:25:eb:d4:8a:30)  
Internet Protocol Version 4, Src: 192.168.1.68, Dst: 52.214.128.28  
Transmission Control Protocol, Src Port: 50962, Dst Port: 443, Seq: 518, Ack: 5381, Len: 126  
Transport Layer Security

- ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 70
  - ▼ Handshake Protocol: Client Key Exchange
    - Handshake Type: Client Key Exchange (16)
    - Length: 66
    - ▼ EC Diffie-Hellman Client Params
      - Pubkey Length: 65
      - Pubkey: 0479558b31c6176b9e6212a67921eb75bb0a0770ee4aa76502d8902504c2096d086052a9...
- ▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  - Content Type: Change Cipher Spec (20)
  - Version: TLS 1.2 (0x0303)
  - Length: 1
  - Change Cipher Spec Message
- ▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 40
  - Handshake Protocol: Encrypted Handshake Message

12. The encryption of the data is then done by the agreed upon cipher from step #9 alone with the public key for the server which is decrypted at the server end via the server's private key.

### Highlighted encrypted data which can be decrypted with server's private key.

64	4.759474	192.168.1.68	52.214.128.28	TLSv1.2	249	Application Data, Application Data, Application Data, Application Data
65	4.764597	192.168.1.68	52.214.128.28	TLSv1.2	779	Application Data

> Frame 64: 249 bytes on wire (1992 bits), 249 bytes captured (1992 bits) on interface \Device\NPF\_{F73D8206-7163-472D-AD5D-8D75AAE4C7C8}, id 0  
> Ethernet II, Src: AzureWav\_e3:71:43 (ec:2e:98:e3:71:43), Dst: Cambridg\_d4:8a:30 (a8:25:eb:d4:8a:30)  
> Internet Protocol Version 4, Src: 192.168.1.68, Dst: 52.214.128.28  
> Transmission Control Protocol, Src Port: 50962, Dst Port: 443, Seq: 644, Ack: 5708, Len: 195  
▼ Transport Layer Security

- ▼ TLSv1.2 Record Layer: Application Data Protocol: http2
  - Content Type: Application Data (23)
  - Version: TLS 1.2 (0x0303)
  - Length: 48
  - Encrypted Application Data: 0000000000000014a9b01d1e99fc5aba25f4ae3d7f94a151715c0f49ec91af3ec7fcc71...
  - [Application Data Protocol: http2]
- ▼ TLSv1.2 Record Layer: Application Data Protocol: http2
  - Content Type: Application Data (23)
  - Version: TLS 1.2 (0x0303)
  - Length: 57
  - Encrypted Application Data: 00000000000000296c7956a5dc48824c304983b9197a76cb619a90dabe4a46297b6bd48...
  - [Application Data Protocol: http2]
- ▼ TLSv1.2 Record Layer: Application Data Protocol: http2
  - Content Type: Application Data (23)
  - Version: TLS 1.2 (0x0303)
  - Length: 37
  - Encrypted Application Data: 000000000000003d15aeebadac8439f7714e028188f40e2a207a86312b20051b8ab945...
  - [Application Data Protocol: http2]
- ▼ TLSv1.2 Record Layer: Application Data Protocol: http2
  - Content Type: Application Data (23)
  - Version: TLS 1.2 (0x0303)
  - Length: 33
  - Encrypted Application Data: 0000000000000041407107dc9a5e14a8fb3fe833afe31f9cc4cb616880e581718...
  - [Application Data Protocol: http2]

13. Once verified an encrypted connection is established between the client and the server, the request is then processed by the server and a response is provided.
14. The response if everything worked correctly, will include a 200 HTTP status code, the content of the website (HTML of the website), headers and other informational data like content size, HTTP Status Description.

```
ubuntu@ip-172-31-86-102:~$ curl https://wettransfer.com -I
HTTP/2 200
date: Fri, 15 Jul 2022 16:09:35 GMT
content-type: text/html; charset=utf-8
cache-control: no-cache, no-store
etag: W/"2bcea18dc5d70a1c1b664a6ec934471c"
expires: Fri, 01 Jan 1990 00:00:00 GMT
pragma: no-cache
referrer-policy: strict-origin-when-cross-origin
set-cookie: _wt_snowplowid.0497=f70c420d-9612-4d63-9c2d-21438d57e
vary: Accept-Encoding
vary: Accept-Encoding, Origin
x-content-type-options: nosniff
x-download-options: noopen
x-frame-options: SAMEORIGIN
x-opaque: c293eaa25aa324648cc6c7fd47396eefe9971b36-k5x66-75096
x-permitted-cross-domain-policies: none
x-request-id: 392183f36efc510e005fcab4c7e8fcee
x-runtime: 0.024516
x-xss-protection: 1; mode=block
strict-transport-security: max-age=15552000; includeSubDomains;

ubuntu@ip-172-31-86-102:~$ █
```

## Answer 2

Please find the required code files and references in [this repo](#).